



TITLE:

論理型言語によるメタ推論とその応用(情報の構造化と意味に関する研究)

AUTHOR(S):

國藤, 進; 北上, 始; 宮地, 泰造; 竹内, 彰一; 横田, 治夫;
古川, 康一; 上田, 和紀

CITATION:

國藤, 進 ...[et al]. 論理型言語によるメタ推論とその応用(情報の構造化と意味に関する研究). 数理解析研究所講究録 1984, 525: 284-291

ISSUE DATE:

1984-06

URL:

<http://hdl.handle.net/2433/98505>

RIGHT:

論理型言語によるメタ推論とその応用

國藤 進(Susumu Kunifuji), 北上 始(Hajime Kitakami),
宮地泰造(Taizo Miyachi), 竹内彰一(Akikazu Takeuchi),
横田治夫(Haruo Yokota), 古川康一(Koichi Furukawa)
((財) 新世代コンピュータ技術開発機構)
上田和紀(Kazunori Ueda) (日本電気(株) C&C研)

1. 序 論

第5世代コンピュータ・プロジェクトにおける研究開発課題である知識ベース管理システムや問題解決・推論システムを構築しようとする際、論理型プログラミング言語Prolog上に知識ベースの更新を管理する機能や一階ホーン論理の枠を超える高次の／高度な(協調)問題解決の機能を実現したいことがある。このような場合、知識ベースに格納されている知識そのものの使い方に関する知識(通常、メタ知識といわれるが、)を用いた推論をも処理対象とするために、メタ推論機構と呼ばれるべき機構の導入を余儀なくされることが多い。

本論文では、そのようなメタ推論の機能を利用可能ならしめるため、一階ホーン論理あるいはその部分集合での証明可能性という概念を抽象化したメタ述語(逐次型Prolog上ではdemo述語、並列型Prolog上ではsimulate述語と呼ばれるが、)を導入する。すなわちここでは、メタ述語demoを用いた逐次メタ推論機能とメタ述語simulateを用いた並列メタ推論機能の導入について考察し、メタ推論の必要性、逐次メタ推論の方式検討とその応用、および並列メタ推論の方式検討とその応用について述べる。

2. メタ推論の必要性

第5世代コンピュータ・システムの最終目標は、知識情報処理システムのプロトタイプ確立である。知識情報処理システム構築のためのガソリンとエンジンとでもいうべき二本柱は、そこで取扱われる知識とその知識を処理する機構である。知識情報処理システム上で実世界に出現する複雑な問題を解決していくには、次に述べる三種類の知識の使い分けが必要である。第一が個別知識(Prologのファクト)として出現する対象データと呼ばれる知識(これ自体、すぐ後に述べる対象知識の一種とみなす

ことも可能であるが、)であり、第二が一般知識(Prologのルール)として出現する対象知識と呼ばれる知識であり、第三が個別知識や一般知識、すなわち対象知識、に関するメタ知識と呼ばれる知識である。これら三種の知識を格納しているデバイスがそれぞれ(対象)データベース、(対象)知識ベース、およびメタ知識ベースである。知識処理機構の立場からみると、これら三種の知識を処理するには、それぞれ高速・大容量の検索機構、推論機構、およびメタ推論機構の整備が不可欠である。システム構築の立場からみると、それぞれデータベース管理システム、知識ベース管理システム、およびメタ知識ベース管理システム(これは従来、推論根拠管理システムといわれていたが、)を論理プログラミング言語Prolog上で構築することを意味する。

1章で述べたように、メタ推論とは「メタ知識を用いた推論」のことである。メタ知識とは「対象知識の使い方に関する知識」、あるいは「(それによって再帰的に定義される)メタ知識の使い方に関する知識」のことである。するとメタ述語demoは逐次型Prolog、第5世代コンピュータ・プロジェクトでは核言語第0版(KL 0)あるいはその部分集合、を対象知識とするホーン論理の問題解決機であり、メタ述語simulateは並列型Prolog、核言語第1版(KL 1)あるいはその部分集合、を対象知識とするホーン論理の問題解決機である。メタ述語demoやsimulateの表現・利用と、逐次型Prologあるいは並列型Prologのインタプリタ/コンパイラの使い方に関する知識の表現・利用とは、等価な概念である。問題解決機としてのdemo述語やsimulate述語を陽に表現・利用することにより、知識ベース管理や問題解決・推論に関連して派生する高度な機能を簡単に実現できる。

メタ知識と対象知識を単一の論理プログラミング言語で表現し、両知識を融合した形式で利用することにより、対象知識の更新や並列プロセスの管理を含む高度なメタ推論機能を簡単に実現できる。逐次型Prologにおいてdemo述語導入の理由は、われわれのグループで研究開発中の大規模関係データベース管理プログラムKAISERに知識獲得機能を付与する際、知識ベースの更新を管理する必要上から生じた、並列型Prologにおいてsimulate述語導入の理由は、論理型言語上でのオペレーティング・システム/プログラミング・システムとユーザ・プログラムとの協調問題解決の必要上から生じた。そこでのインプリメンテーション経験を踏まえて要約すれば、demo述語やsimulate述語というメタ推論機能を導入せざるを得なかった必然性は、次のような諸点にある。

- (a) 複数の対象知識、あるいはそれらから構成される構造をもった知識を、論理型言語上でいかに取扱うべきか。
- (b) 与えられた対象知識あるいはメタ知識上で与えられたゴールを解く過程の全体、すなわち失敗/成功の全履歴を含むproof tree、をいかに析出するか。このproof treeを適宜変換することにより、与えられた問題を解くのに必要なメタ情

報が抽出できる。

- (c) 与えられた対象知識やメタ知識の下で、与えられたゴールを解く評価のモードや探索空間の探索の仕方を制御するにはどのようにしたらよいか。
- (d) 対象知識とメタ知識の間の論理的関係を、互いに閉じたものとして構築するにはどのようにしたらよいか。

以上述べてきた要求を満足するメタ推論用プリミティブとして提案したのが、3章で述べるdemo述語と4章で述べるsimulate述語である。

3. 逐次メタ推論の方式検討とその応用

demo述語に基づく逐次メタ推論を実行するために、知識の表現や利用の対象として、いる対象知識の世界、すなわち対象世界、とそれを操作対象として管理しているメタ知識の世界、すなわちメタ世界、とをそれぞれ論理的に閉じた世界として構築する。ここに論理的に閉じているとは、他の世界との情報交換が特定のメタ情報抽出用論理変数に対するユニフィケーション操作によってしか行われないこと、および対象世界のfailがメタ世界のfailに波及しないことを意味する。これによって、ある世界の内と外との間の論理的つながりを、特定の論理変数を除いて、切断することができる。

逐次メタ推論方式として著者らの提案している最も一般的な逐次メタ推論用demo述語は、次のような形式をしている。

① demo(World, Goal, Result, Control)

このdemo述語は、ある世界Worldにおいて、与えられた制御情報Controlに従って、与えられたゴールGoalを証明していくプロセスを実際に示し(demonstrate)、その証明のプロセスから必要なメタ情報Resultを抽出する。

上述のdemo述語①に対して、次のような省略した形式のdemo述語が、しばしばある種の応用では有効である。その理由は、主としてそれらのインプリメンテーションの容易さとパフォーマンスの向上のためである。

② demo(World, Goal, Result)

③ demo(World, Goal)

なお対象世界として、逐次型Prologの内部データベース自身を用いる場合は、①～③の第1引数Worldは省略できる。③の第1引数を省略したメタ述語“demo(Goal)”

が“Prolog interpreter in Prolog”であり、マニュアル等で知られている。

demo述語①～③をプリミティブとする逐次メタ推論方式の応用例として、既に次のような研究課題が検討された。その詳細については、引用文献を参照されたい。

- (a) 知識獲得システムにおける知識同化機構の実現 [Miyachi 83-1, 83-2, 84-1, 84-2, Kunifuji 83-2, Kitakami 83-2]
- (b) 知識獲得システムにおいて例外値を含む知識同化機構の実現 [Kunifuji 83-3]
- (c) 知識獲得システムにおける知識調節機構の実現 [Kitakami 83-1, 83-2, 84]
- (d) 逐次型推論マシンと関係データベースマシンのインタフェース機構の実現 [Kunifuji 82, 83-1 Yokota 83-1, 83-2, 84]

並列メタ推論の方式検討とその応用

simulate述語に基づくメタ推論を実行するため、知識表現の対象としている対象知識の世界、すなわち対象世界、とそれを操作対象とし管理しているメタ知識の世界、すなわちメタ世界、とをそれぞれ論理的に閉じた世界として構築する。ここに、論理的に閉じているとは、他の世界との情報交換が、特定の通信用チャネルに対するユニフィケーション操作によってしか行われないこと、および対象世界のfailがメタ世界のfailに波及しないことを意味する。これによって、ある世界の内と外との間の論理的つながりを、通信用論理変数を除いて、切断することができる。

並列メタ推論方式として著者らの提案している最も一般的な並列メタ推論用simulate述語は、次のような形式をしている [Kunifuji 84]

④ simulate(World, NewWorld, Goal, Result, Control, Channel)

このsimulate述語は、ある世界Worldにおいて、他の世界と通信チャネルChannelを通じてメッセージ交換しながら、与えられたゴールGoalを制御条件Control 下で解くプロセスを模倣する。模倣の結果、世界Worldは新世界NewWorldに更新されると同時に、ゴールを証明していくプロセスそのものから、必要なメタ情報Resultが順次抽出されていく。

上述のsimulate述語④に対して、次のような省略した形式のものが、しばしばある種の応用では有効である。

⑤ simulate(World, NewWorld, Goal, Result, Channel)

- ⑥ `simulate(World, NewWorld, Goal, Channel)`
- ⑦ `simulate(World, NewWorld, Goal)`
- ⑧ `simulate(World, Goal)`

なお、対象世界として、内部データベース自身を用いる場合は、④～⑧の第1引数 `World` は省略できる。この場合“KL1 interpreter in KL1”という概念に相当する。

`simulate`述語⑤～⑧をプリミティブとする並列メタ推論方式の応用例として、既に次のような研究課題が検討された。その詳細については、引用文献を参照されたい。

(a) 知識プログラミング言語／システムの実現

[Furukawa 83, 84-1, 84-2, Takeuchi 84]

(b) システム・プログラミングへの適用 [Shapiro 84]

`simulate`述語のその他の応用例として、並列プロセスの編集／虫取り、並列現象のシミュレーション、および並列型協調問題解決、等が考察中である。

5. 結 論

論理型言語Prolog上に逐次メタ推論を行うメタ述語demoと並列メタ推論を行うメタ述語`simulate`を導入し、その方式検討と応用可能性について報告した。ここで述べられた応用例以外に、次のような応用例システムが、既に他の著者らによって研究開発中である。

(a) ルールの適用可能性を調べる知識表現言語の試作 [Mizoguchi 84]

(b) はい／いいえの理由を説明するProlog処理系の作成 [Walker 83]

(c) 知識ベース・エディタの実現 [Takeda 83]

前章までに述べてきたことおよび上述の研究成果等を踏まえて、著者らは更に新たな応用システムの研究開発を目ざしている。著者らの当面の検討課題は、いかに高速なメタ推論機構をもつProlog処理系を実現するかである。これについては論理型言語によるモジュラー・プログラミング技法導入の枠の中で詳細に検討中である。

[引用文献]

- [Furukawa 83] Furukawa, K., Takeuchi, A., and Kunifuji, S., Mandala : A Knowledge Representation System in Concurrent Prolog, Information Society of Japan, Preprints of WG on Knowledge Engineering and Artificial Intelligence, Nov. 1983.
- [Furukawa 84-1] 古川康一, 竹内彰一, 國藤 進, Mandala : A unified system for modular programming and knowledge representation in Concurrent Prolog, 京都大学数理解析研究所, 「情報の構造化と意味に関する研究」研究集会, 1984年 2月.
- [Furukawa 84-2] 古川康一, 竹内彰一, 國藤 進, Mandala : Concurrent Prolog 上の知識プログラミング言語/システム, 情報処理学会第28回全国大会, 電気通信大学, 1984年 3月.
- [Kitakami 83-1] 北上 始, 麻生盛敏, 國藤 進, 宮地泰造, 古川康一, 知識同化機構の一実現法, 情報処理学会知識工学と人工知能研究会資料 30-2, 1983 年 6月(TR-010)
- [Kitakami 83-2] 北上 始, 宮地泰造, 國藤 進, 古川康一, 知識獲得システムの一実現法, 情報処理学会第27回全国大会, 名古屋大学, 1983年10月(TM-0017).
- [Kitakami 84] Kitakami, H., Kunifuji, S., Miyachi, T., and Furukawa, K., A Methodology for Implementation of A Knowledge Acquisition System, Proc. of the 1984 International Symposium on Logic Programming, Atlantic City, U.S.A., Feb. 6-9, 1984(TR-037).
- [Kunifuji 82] Kunifuji, S. and Yokota, H., PROLOG and Relational Data Bases for Fifth Generation Computer Systems, Proc. of CERT Workshop on "Logical Base for Data bases", DEC. 1982(TR-002).
- [Kunifuji 83-1] 國藤 進, 横田治夫, 古川康一, 北上 始, 宮地泰造, 論理型言語と関係データベース管理システムとのインタフェース(1) 基本的考え方, 情報処理学会第26回全国大会, 東京工業大学, 1983年 3月.

- [Kunifuji 83-2] 國藤 進, 麻生盛敏, 竹内彰一, 宮地泰造, 北上 始, 横田治夫, 安川秀樹, 古川康一, Prologによる対象知識とメタ知識の融合とその応用, 情報処理学会知識工学と人工知能研究会資料30-1, 1983年 6月(TR-009).
- [Kunifuji 83-3] 國藤 進, 宮地泰造, 北上 始, 古川康一, 知識獲得とメタ推論, 情報処理学会第27回全国大会, 名古屋大学, 1983年10月(TM-0016).
- [Kunifuji 84] 國藤 進, 竹内彰一, 古川康一, 上田和紀, メタ推論とその応用—並列メタ推論用メタ述語simulateについて—, 情報処理学会第28回全国大会, 電気通信大学, 1984年 3月(TM-0039).
- [Miyachi 83-1] 宮地泰造, 國藤 進, 北上 始, 古川康一, 竹内彰一, 横田治夫, 論理データベース向きの知識同化方式の一提案, 京都大学数理解析研究所, 「モデル表現とその構築に関する理論と実際の研究」研究集会, 1983年 2月(TM-0004).
- [Miyachi 83-2] 宮地泰造, 北上 始, 國藤 進, 古川康一, 竹内彰一, 横田治夫, KAISERにおける知識獲得について, 情報処理学会第26回全国大会, 東京工業大学, 1983年 3月.
- [Miyachi 84-1] Miyachi, T., Kunifuji, S., Kitakami, H., Furukawa, K., Takeuchi, A. and Yokota, H., A Knowledge Assimilation Method for Logic Databases, Proc. of the 1984 International Symposium on Logic Programming, Atlantic City, U.S.A., Feb. 6-9, 1984.
- [Miyachi 84-2] 宮地泰造, 國藤 進, 北上 始, 古川康一, 知識獲得システムにおける一貫性保持について, 情報処理学会第28回全国大会, 電気通信大学, 1984年 3月(TM-0040).
- [Mizoguchi 84] 溝口文雄, 本田栄司, 片山佳則, オブジェクト指向概念を導入した知識表現言語: 姿と鏡の設計とその応用, Proc. of the Logic Programming Conference '84, 東京, March. 1984.
- [Shapiro 84] Shapiro, E., Systems Programming in Concurrent Prolog, 11th Annual ACM SIGACT/SIGPLAN Symposium on Principles of Programming Languages, Salt Lake City, U.S.A., Jan. 1984.

- [Takeda 83] 武田正之, 知識ベースに基づくLanguage-oriented Editorの設計,
Workshop of ICOT WG4, July 1983.
- [Takeuchi 84] 竹内彰一, 古川康一, 國藤 進, Mandala : Concurrent Prolog 上の
知識プログラミング言語/システム-その2 Implementation -, 情報処理学
会第28回全国大会, 電気通信大学, 1984年 3月.
- [Walker 83] Walker, A., Prolog/Ex1, An Inference Engine which Explains both
Yes and No Answers, Proc. of IJCAI 83, Karlsruhe, West Germany,
Aug. 1983.
- [Yokota 83-1] 横田治夫, 國藤 進, 古川康一, 北上 始, 宮地泰造, 論理型言語と
関係データベース管理システムとのインタフェース(2) インプリメンテーショ
ン, 情報処理学会第26回全国大会, 東京工業大学, 1983年 3月.
- [Yokota 83-2] 横田治夫, 國藤 進, 柴山茂樹, 宮崎収兄, 角田健男, 村上国男,
Prologによる推論機構と関係データベースの結合, 情報処理学会知識工学と人
工知能研究会資料 32-2, 1983年11月.
- [Yokota 84] Yokota, H., Kunifuji, S., Kakuta, T., Miyazaki, N., Shibayama, S.,
and Murakami, K., An Enhanced Inference Mechanism for Generating
Relational Algebra Queries, Proc. of Third ACM SIGACT-SIGMOD Symposium
on Principles of Database Systems, Waterloo, Canada, April 2-4, 1984.